(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 August 2001 (23.08.2001)

PCT

(10) International Publication Number
WO 01/61504 A1

(51) International Patent Classification⁷: G06F 12/14

(21) International Application Number: PCT/US01/04583

(22) International Filing Date: 12 February 2001 (12.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/503,878 14 February 2000 (14.02.2000) US

(71) Applicant: WILDSEED, LTD. [US/US]; 550 Kirkland Way N.E., Suite 100, Kirkland, WA 98033 (US).

(72) Inventor: PORTER, Swain, W.; 12511 89th Court, N.E., Kirkland, WA 98034 (US).

(74) Agents: AUYEUNG, Aloysius, T., C. et al.; Columbia IP Law Group, LLC, Suite 109, 4900 SW Meadows Road, Lake Oswego, OR 97035 (US).

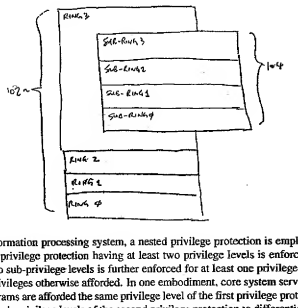
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: PROTECTIVELY OPERATING A PROCESSING DEVICE WITH A MAIN PROTECTION RING HAVING AT MAIN-LEVELS SUB-RING LEVELS



(57) Abstract: In a data/information processing system, a nested privilege protection is employed to protect the system when executing instructions. A first privilege protection having at least two privilege levels is enforced. Additionally, a second privilege protection having at least two sub-privilege levels is further enforced for at least one privilege level of the first privilege protection to further differentiate the privileges otherwise afforded. In one embodiment, core system services, programming language runtime support and application programs are afforded the same privilege level of the first privilege protection, and the different types of programs are afforded different sub-privilege levels of the second privilege protection to differentiate the privileges afforded by the first privilege protection. In one embodiment, the differential sub-privilege level protection is further extended to application programs of different sources, making the sytem particularly suitable for networked applications, such as accessing web servers on the Internet. In one embodiment, the first privilege protection is hardware facilitated, while the second privilege protection is software facilitated.

WO 01/61504 A1

PROTECTIVELY OPERATING A PROCESSING DEVICE WITH A MAIN PROTECTION RING HAVING AT MAIN-LEVELS SUB-RING LEVELS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of electronic data/information processing. More specifically, the present invention relates to methods and apparatuses for protectively operating data/information processing devices.

2. Background Information

The term "data/information processing devices" as used herein is intended to include all microprocessor based devices and/or systems, operated under the control of an operating system. Examples of these devices/systems include but are not limited to general as well as special purpose computing devices/systems, regardless of form factors, palm sized, laptops, desktops, rack mounted, and the like. Examples of special purpose computing devices include but are not limited to set-top boxes, wireless communication devices, and the like. The term "operating system" as used herein is intended to include all software provided to manage and facilitate application usage of hardware resources, however minimal the control and resource scope may be. Typical resource management functions of an "operating system" include task scheduling, memory management and the like. The term "task" as used herein is intended to include its common meaning of an executing instance of a program (a collection of programming instructions).

Ever since the early days of computing, computer systems have provided privilege protection to protect the system from being brought down by failures of non-essential programs, such as application programs. The IBM 360 systems provided a supervisor mode and a user mode to segregate privileged system

programs and unprivileged user programs. The Multics (Multiplexed Information and Computing Service) developed by Massachusetts Institute of Technology, in cooperation with others, employed a 64 ring approach, combining access node and a triple of ring numbers (r1, r2, r3). In U.S. Patent 4,177,510, issued to Appell et al., a hardware facilitated 4 ring approach is disclosed. Today, the Intel Architecture processors are known to provide a 4 ring hardware facilitated protection through the employment of memory segment descriptors and current task privilege level (CPL). However, partly because most of the other microprocessors remain having a two mode protection approach, the Windows® operating system, used in most Intel Architecture compatible processors, merely employ two of the four ring protection provided by the hardware. The virtual memory manager and various virtual device drivers (VxD) are executed in ring 0, while all other programs, including kernel system services and so forth are executed out of ring 3.

The two levels of protection were reasonably adequate in the days when few programs are executed on most computer systems. Moreover, most of the computer systems operate by themselves, with few interactions from the outside world.

Advances in microprocessor, telecommunication and networking technology have dramatically expanded the applications of computing devices, and changed their operating environment. Today, most data/information processing systems are connected to private and/or public networks, such as the Internet, executing programs that are dynamically downloaded from a number of sources. Some sources are trustworthy, and their programs tend to be well behaved, but others are not.

Accordingly, a need exists to improve the protection of data/information processing systems.

SUMMARY OF THE INVENTION

In a data/information processing system, a nested privilege protection is employed to protect the system when executing instructions. A first privilege protection having at least two privilege levels is enforced. Additionally, a second privilege protection having at least two sub-privilege levels is further enforced for at least one privilege level of the first privilege protection to further differentiate the privileges otherwise afforded.

In one embodiment, core system services, programming language runtime support and application programs are afforded the same privilege level of the first privilege protection, and the different types of programs are afforded different sub-privilege levels of the second privilege protection to differentiate the privileges afforded by the first privilege protection. In one embodiment, the differential sub-privilege level protection is further extended to application programs of different sources, such as trusted and untrusted.

In one embodiment, the first privilege protection is hardware facilitated, while the second privilege protection is software facilitated.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

Figure 1 illustrates an overview of the present invention, in accordance with one embodiment;

Figure 2 illustrates a method of the present invention in accordance with one embodiment;

Figure 3 illustrates an example data/information processing system suitable for practicing the present invention; and

Figure 4 illustrates the relevant enhancement to the operating system of **Fig. 3**.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, various aspects of the present invention will be described. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, it will also be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented using terms such as scripts, applet, end-user interfaces, icons, and so forth, commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. Parts of the description will be presented in terms of operations performed by a computer system, using terms such as registering, notifying, sending, and so forth. As well understood by those skilled in the art, these quantities and operations take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and otherwise manipulated through mechanical and electrical components of a digital system; and the term digital system include general purpose as well as special purpose data processing machines, systems, and the like, that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps performed in turn in a manner that is most helpful in understanding the present invention, however, the order of description should not be construed as to imply that these operations are necessarily order dependent, in particular, the order the steps are presented. Furthermore, the phrase "in one embodiment" will be used repeatedly,

however the phrase does not necessarily refer to the same embodiment, although it may.

Referring now to **Figure 1**, wherein an overview of the present invention in accordance with one embodiment is shown. As illustrated, in accordance with the present invention, a nested privilege protection approach is employed to protectively operate a data/information processing system (hereinafter, simply system). Nested privilege protection **100** includes at least two substantially independent privilege protections to protect the system during instruction execution (i.e. from their failures). For the illustrated embodiment, two such privilege protections **102** and **104** are shown. First privilege protection **102** has at least two privilege levels enforced at the system level for executing instructions. For the illustrated embodiment, four privilege levels are shown, ring 0 through ring 3. Second privilege protection **104** also has at least two sub-privilege levels enforced for at least one of the privilege level for executing instructions, to further differentiate the privileges afforded to different executing programs otherwise accorded by first privilege protection **102**. For the illustrated embodiment, four sub-privilege levels, sub-ring 0 through sub-ring 3, for ring 3 of first privilege protection **102** are shown. As a result, system protection is advantageously strengthened over the protection otherwise afforded by employing only first privilege protection **102**. In particular, when the privilege protection afforded by first privilege protection **102** is not fully taken advantage (as in the case of Window® and Intel Architecture processors), the present invention advantageously enables the protection sacrificed to be recovered.

The term "privilege ring" or "ring" as used herein is intended to include its conventional meaning that a program afforded a more inner privilege ring (or sub-ring) typically has privileges inclusive that of another program afforded a more outer privilege ring (or sub-ring). The details of these privileges (and their

further differentiation) are implementation dependent, and non-essential to the understanding of the present invention.

Before further describing the present invention, it should also be noted that while four rings are shown for first privilege protection **102**, the present invention may be practiced with first privilege protection **102** enforcing less or more privilege levels. Similarly, while second privilege protection **104** is shown for only one privilege level of first privilege protection **102**, and having four sub-privilege level, the present invention may be practiced with second privilege protection **104** being employed for more than one privilege level of first privilege protection **102**, and/or having less or more sub-privilege levels. Likewise, additional nested sub-privilege protection may be employed (e.g. for one of the sub-privilege level of second privilege protection **104**) to further differentiate the privileges otherwise afforded by first and second privilege protections **102** and **104**.

Referring now to **Figure 2**, wherein an example application of the present invention in accordance with one embodiment is shown. For the illustrated embodiment, second privilege protection **104'** having four sub-privilege levels, sub-ring 0 through sub-ring 3, is employed to provide further protection to a system by affording additional differentials to the privileges of core system services **152**, programming language runtime supports **154** for different programming languages, application programs of a first kind **156**, and application programs of a second kind **158**, which are otherwise accorded the same privileges under first privilege protection **102'** (i.e. being assigned the same privilege ring, e.g. ring 3). More specifically, core system services **152** are afforded the full privileges otherwise accorded, programming language runtime supports **154** are afforded a first smaller subset of these otherwise accorded privileges, application programs of the first kind **156** are afforded a second yet even smaller subset, and finally, application programs of the second kind **158** are

afforded a third yet even smaller (e.g. minimal) subset. The privilege boundaries of these subsets are application dependent. Thus, as alluded to earlier, the illustrated embodiment enables additional protection to be afforded, or at a minimum, recovering protections of first privilege protection **102'** otherwise sacrificed.

What constitutes application programs of a first kind **156** versus a second kind **158** is also application dependent. In one implementation, program codes, such as scripts and applets, dynamically downloaded by a browser (e.g. from various Web servers through the Internet) are considered as application programs of the second kind **158**, to be afforded the least privileged sub-privilege ring. Thus, it can be seen a system incorporated with the present invention is particularly suitable for networked systems where the systems are exposed to programs from a variety of uncertain sources.

In alternate embodiments, other differentiation of the otherwise afforded privileges may be made instead. For example, instead of according all scripts and applets dynamically downloaded by the browser the least privileged sub-privilege ring, scripts and applets dynamically downloaded from a trustworthy site (such as, an intranet web site of a corporation) may nevertheless be considered applications of the first kind and accorded privileges of the next more privileged privilege ring. Similarly, instead of according all programming language runtime supports **154** the same sub-privilege ring, runtime supports for some programming languages may be afforded a more privileged ring, while runtime supports for other programming languages may be afforded a lesser privilege ring.

Referring now to **Figure 3**, wherein a block diagram illustrating an example system incorporated with the teachings of the present invention is shown. As shown, example system **200** includes processor **202**, ROM **203**, and system memory **204** coupled to each other via "bus" **206**. Coupled also to "bus"

206 are non-volatile mass storage 208, display device 210, cursor control device 212 and communication interface 214. ROM 203 includes a basic input/output system (BIOS) 205. During operation, memory 204 includes working copies of applications 224, programming language runtime supports 223, and working copies of operating system 222. In one embodiment, either applications 224 or operating system 222 includes a browser (not shown) for accessing local as well as remote information, such as web pages from various web servers available on an intranet or the Internet.

Processor 202 is equipped with hardware support to implement first privilege protection 102. Operating system 222 is incorporated with the teachings of the present invention, implementing second privilege protection 104, to be described in more detail below. In other words, for the illustrated embodiment, first privilege protection 102 is hardware facilitated, while second privilege protection 104 is software facilitated. In alternate embodiments, both protections may be hardware or software facilitated.

Examples of processors 202 include but are not limited to processors of the Pentium® family available from Intel Corporation of Santa Clara, CA, and processors of the PowerPC® family available from IBM of Armonk, NY. Except for the teachings of the present invention incorporated, operating system is otherwise intended to represent a wide range of operating systems known in the art. Examples of operating system 222 that may be enhanced include but are not limited to the Window® operating system available from Microsoft Corp., of Redmond, WA, and the Linux operating system, available e.g. from Red Hat Inc. of Durham, NC.

Similarly, each of the other enumerated elements is intended to represent a wide range of the respective devices/elements known in the art. For example, ROM 203 may be EEPROM, Flash and the like; and memory 204 may be SDRAM, DRAM and the like, from semiconductor manufacturers such as Micron Technology of Boise, Idaho. Bus 206 may be a single bus or a multiple bus

implementation. In other words, bus **206** may include multiple buses of identical or different kinds properly bridged, such as Local Bus, VESA, ISA, EISA, PCI and the like. Mass storage **208** may be disk drives or CDROMs from manufacturers such as Seagate Technology of Santa Cruz of CA, and the like. Typically, mass storage **208** includes the permanent copy of operating system **222**, runtime support **223**, and some applications **224**. The permanent copy may be installed in the factory, or in the field. For field installation, the permanent copy may be distributed using article of manufactures with recordable medium such as diskettes, CDROM, DVD and the like, or downloaded from a distribution server through a data network (such as the Internet). The distribution server may be a server of the OEM, i.e. the software developer. Display device **210** may be monitors of any types from manufacturers such as Viewsonic of Walnut, CA. Cursor control **212** may be a mouse, a track ball and the like, from manufacturers such as Logictech of Milpitas, CA. Communication interface **214** may be a modem interface, an ISDN adapter, a DSL interface, an Ethernet or Token ring network interface and the like, from manufacturers such as 3COM of San Jose, CA.

Before further describing the enhancements made to operating system **222**, it should be noted the present invention may also be practiced without some of the enumerated elements, e.g. mass storage **208**, or with additional elements, such as graphics accelerators, audio and video add-on cards, and so forth.

Referring now to **Figure 4**, a block diagram illustrating the enhancements made to operating system **222** in accordance with one embodiment is shown. For ease of understanding, only the elements of operating system **222** relevant to the understanding of the present invention are shown. As illustrated, for the embodiment, operating system **222** includes first and second memory mangers **252** and **254**, and program loader **256**. In one embodiment, program loader **256** is part of core system services **152**. First memory manager **252** by design

accords itself a first higher privilege level of first privilege protection **102** (e.g. ring 0) facilitated by processor **202**, and core system services **152**, runtime supports **154** and application programs **156-158** a second lower privilege level of first privilege protection **102** (e.g. ring 3) facilitated by processor **202**. In one embodiment, first memory manager **252** accords the privileges by setting a current task privilege level (CPL) when a program of the various types **152-158** is invoked for execution. As alluded to earlier, the privilege is enforced by processor **202** in accordance with the CPL. First memory manager **252** is known in the art, thus will not be further described. For additional information on memory descriptors and CPL, see e.g. product literatures for Intel Architecture processors.

Program loader **256** is enhanced to modify each program invoked, including in particular, a memory allocation request service program (not shown) of core system services **152**, to afford second memory manager **254** the opportunity to enforce second privilege protection **104**. In one embodiment, the modifications include modifying the memory allocation request service program to trap all memory allocation requests from executing tasks to second memory manager **254** for processing. For the embodiment, the modifications further include modifying each program being invoked for execution such that all memory references will be re-routed to second memory manager **254** for processing. These modifications include modifying all Load and Store instructions with indirect Load and Stores where the load and store addresses are to be obtained from an address table (not shown) under the control of second memory manager **254**, including insertion of additional instructions where necessary. For example, in the case of computing A = B + C and then Load A (A being an address), the modifications include inserting a store to store address A into the address table and modifying the Load instruction to Load @ptr (where ptr points to the offset in the address table where address A is stored). These and other ancillary modifications are similar to the techniques employed by compilers

in handling register allocations. They are well within the abilities of those ordinarily skilled in the art, accordingly will not be each individually described.

Second memory manager **254** is provided to facilitate and administer second privilege protection **104**. By virtue of modification to the memory allocation request service program, second memory manager **254** is by design in charge of all memory allocation for programs to be executed with the second lower privilege level of first privilege protection **102**. In one embodiment, where second memory manager **254** accords four sub-privilege levels, second memory manager **254** first obtains a memory pool from first memory manager **252**, and services memory requests from core system services **152** by allocating memory locations from a first sub-pool having first n common lower order bits, memory requests from runtime supports **154** by allocating memory locations from a second sub-pool having second n common lower order bits, memory requests from application programs **156** by allocating memory manager **254** locations from a third sub-pool having third n common lower order bits, and memory requests from runtime supports **158** by allocating memory locations from a fourth sub-pool having fourth n common lower order bits (see **Fig. 5**). For example, from a 1MB memory pool obtained from first memory manager **252**, second memory manager **254** services memory requests from core system services **152** by allocating memory locations from a first 256K sub-pool constituted with the lowest 1K locations of each 4K page, memory requests from runtime supports **154** by allocating memory locations from a second 256K sub-pool constituted with the second lowest 1K locations of each 4K page, and so forth. As a result, the various programs accorded the same privilege level protection of the first privilege protection, but different sub-privilege levels are separated from each other.

Similarly, by virtue of modifications to re-route all memory references of programs accorded the second lower privilege level of first privilege protection **102**, second memory manager **254** is enabled to examine each memory

reference made by these programs, to ensure in substance they are referencing only memory locations that are within their privilege scopes. Note that for the illustrated embodiment, it is not necessary for second memory manager **254** to fully resolve that a memory reference is definitively within the referencing program's privilege scope. It is suffice for second memory manager **254** to ensure that if the memory reference is referencing the memory pool managed by second memory manager **254**, the reference is within the scope of the sub-privilege level, as improper reference to other memory locations not managed by second memory manager **254** will be protected by first privilege protection **102**.

Thus, a method and an apparatus for protectively operating a data/information processing system has been described. While the present invention has been described in terms of the above illustrated embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The present invention can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of restrictive on the present invention.

CLAIMS

What is claimed is:

1. An apparatus comprising:

one or more memory modules;

a processor coupled to the one or more memory modules to execute programming instructions stored in said memory module(s), the processor being equipped to provide a first hardware facilitated privilege protection for programming instructions to be executed, with at least two privilege levels; and

a storage medium having stored therein a first and a second plurality of programming instructions implementing a first and a second memory manager to manage usage of said memory, with the first memory manager managing the hardware facilitated privilege level to be afforded by the processor for different sets of programming instructions to be executed, and the second memory manager managing and further facilitating a second software facilitated privilege protection for the different sets of programming instructions to be executed for at least one of said hardware facilitated privilege level, the second software facilitated privilege protection having also at least two sub-privilege levels.

2. The apparatus of claim 1, wherein the first memory manager is equipped to cause the processor to afford itself a first hardware facilitated privilege level, and the second memory manger a second hardware facilitated privilege level having less or equal privileges than the first hardware facilitated privilege level.

3. The apparatus of claim 1, wherein the second memory manger is equipped to afford itself a first software facilitated sub-privilege level having most privileges than all other software facilitated sub-privilege levels.

4. The apparatus of claim 1, wherein the first memory manager is equipped to further cause the processor to afford the same hardware facilitated privilege level to a first plurality of sets of programming instructions for providing core system services, and to a second plurality of sets of programming instructions for providing programming language runtime support for programming instructions of a plurality of programming languages, and the second memory manger is equipped to afford a first software facilitated sub-privilege level to the core system services programming instruction sets, and a second software facilitated sub-privilege level to the programming language runtime support sets, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

5. The apparatus of claim 1, wherein the first memory manager is further equipped to cause the processor to afford the same hardware facilitated privilege level to a first plurality of sets of programming instructions for providing programming language runtime support for programming instructions of a plurality of programming languages, and to a second plurality of sets of programming instructions of said programming languages implementing application functions, and the second memory manger is equipped to afford a first software facilitated sub-privilege level to the programming language runtime support sets, and a second software facilitated sub-privilege level to the application function implementation sets, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

6. The apparatus of claim 1, wherein the first memory manager is further equipped to cause the processor to afford the same hardware facilitated privilege level to a first and a second plurality of sets of programming instructions for providing programming language runtime support for programming instructions of

a first and a second programming language, and the second memory manger is equipped to afford a first and a second software facilitated sub-privilege level to the first and second programming language runtime support sets respectively, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

7. The apparatus of claim 6, wherein the first programming language runtime support sets are from a first source, and the second programming language runtime support sets are from a second source.

8. The apparatus of claim 1, wherein the first memory manager is further equipped to cause the processor to afford the same hardware facilitated privilege level to a first and a second plurality of sets of application implementing programming instructions, and the second memory manger is equipped to afford a first and a second software facilitated sub-privilege level to the first and second application implementing sets respectively, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

9. The apparatus of claim 8, wherein the first application implementing sets are from a trusted source, and the second application implementing sets are from an untrusted source.

10. The apparatus of claim 8, wherein the first application implementing sets are web pages downloaded from a first web server, and the second application implementing sets are web pages downloaded from a second web server.

11. The apparatus of claim 1, wherein the storage medium further having stored therein a third plurality of programming instructions implementing a

program loader equipped to fix up each set of programming instructions it loads for execution by the processor to afford the second memory manager the opportunity to administer the second software facilitated privilege protection.

12. The apparatus of claim 11, wherein the program loader is equipped to fix up a memory allocation request service program to trap each memory allocation request by a programming instruction set to the second memory manager.

13. The apparatus of claim 12, wherein the second memory manager is equipped to allocate memory in response to memory requests from programming instruction sets to be afforded the same second software facilitated sub-privilege level from a sub-pool of memory locations with the same n lower order bits of a memory pool, where n is an integer.

14. The apparatus of claim 11, wherein the program loader is equipped to fix up each set of programming instructions it loads for execution by the processor to re-route each memory reference by the programming instruction set through the second memory manager.

15. The apparatus of claim 14, wherein the second memory manager is equipped to validate all memory references made by programming instruction sets to be afforded the same second software facilitated sub-privilege level reference only an eligible sub-pool of memory locations with the same n lower order bits of a memory pool, where n is an integer.

16. A method comprising:

enforcing a first privilege protection with at least two privilege levels through hardware facilitated privilege protection when executing programming instructions; and

concurrently, enforcing a second privilege protection with at least two sub-privilege levels through software facilitated privilege protection for at least one of the hardware facilitated privilege level.

17. The method of claim 16, wherein said enforcing of hardware facilitated privilege protection comprises affording a first memory manager a first hardware facilitated privilege level, and a second memory manger a second hardware facilitated privilege level having less or equal privileges than the first hardware facilitated privilege level.

18. The method of claim 17, wherein said enforcing of software facilitated privilege protection comprises affording the second memory manger a first software facilitated sub-privilege level having most privileges than all other software facilitated sub-privilege levels.

19. The method of claim 16, wherein

said enforcing of hardware facilitated privilege protection further comprises affording the same hardware facilitated privilege level to a first plurality of sets of programming instructions for providing core system services, and to a second plurality of sets of programming instructions for providing programming language runtime support for programming instructions of a plurality of programming languages; and

said enforcing of software facilitated privilege protection comprises affording a first software facilitated sub-privilege level to the core system services programming instruction sets, and a second software facilitated sub-privilege level to the programming language runtime support sets, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

20. The method of claim 16, wherein

said enforcing of hardware facilitated privilege protection further comprises affording the same hardware facilitated privilege level to a first plurality of sets of programming instructions for providing programming language runtime support for programming instructions of a plurality of programming languages, and to a second plurality of sets of programming instructions of said programming languages implementing application functions; and

said enforcing of software facilitated privilege protection comprises affording a first software facilitated sub-privilege level to the programming language runtime support sets, and a second software facilitated sub-privilege level to the application function implementation sets, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

21. The method of claim 16, wherein

said enforcing of software facilitated privilege protection further comprises affording the same hardware facilitated privilege level to a first and a second plurality of sets of programming instructions for providing programming language runtime support for programming instructions of a first and a second programming language; and

said enforcing of software facilitated privilege protection comprises affording a first and a second software facilitated sub-privilege level to the first and second programming language runtime support sets respectively, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

22. The method of claim 21, wherein the first programming language runtime support sets are from a first source, and the second programming language runtime support sets are from a second source.

23. The method of claim 16, wherein

said enforcing of software facilitated privilege protection further comprises affording the same hardware facilitated privilege level to a first and a second plurality of sets of application implementing programming instructions; and

said enforcing of software facilitated privilege protection comprises affording a first and a second software facilitated sub-privilege level to the first and second application implementing sets respectively, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

24. The method of claim 23, wherein the first application implementing sets are from a trusted source, and the second application implementing sets are from an untrusted source.

25. The method of claim 23, wherein the first application implementing sets are web pages downloaded from a first web server, and the second application implementing sets are web pages downloaded from a second web server.

26. The method of claim 16, wherein said method further comprises fixing up each set of programming instructions being loaded for execution to afford the opportunity to enforce the second software facilitated privilege protection.

27. The method of claim 26, wherein said fixing up comprises fixing up a memory allocation request service program to trap each memory allocation request by a programming instruction set to facilitate service of the memory allocation request in a manner instrumental to the performance of said enforcing of the second software facilitated privilege protection.

28. The method of claim 27, wherein the method further comprises allocating memory in response to memory requests from programming instruction sets to be afforded the same second software facilitated sub-privilege level from a sub-pool of memory locations with the same n lower order bits of a memory pool, where n is an integer.

29. The method of claim 26, wherein said fixing up comprises fixing up each set of programming instructions being loaded for execution to re-route each memory reference by the programming instruction set for validity checking.

30. The method of claim 29, wherein the method further comprises validating all memory references made by programming instruction sets to be afforded the same second software facilitated sub-privilege level reference only an eligible sub-pool of memory locations with the same n lower order bits of a memory pool, where n is an integer.

31. An apparatus comprising:

first means for enforcing a first privilege protection with at least two privilege levels when executing programming instructions; and

second means for concurrently, enforcing a second privilege protection with at least two sub-privilege levels for at least one of the privilege level of the first privilege protection.

32. The apparatus of claim 31, wherein

said first means is equipped to cause the same privilege level of the first privilege protection to be afforded to a first and a second plurality of sets of programming instructions of a first and a second type, and

said second means is equipped to afford a first and a second sub-privilege level of the second privilege protection to the first and second programming

instruction sets of the first and second types respectively, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

33. The apparatus of claim 32, wherein the first and second types are two selected from a group consisting of core system services, programming language runtime supports, and application functions.

34. The apparatus of claim 32, wherein the first and second types are selected from a group consisting of programming language runtime support of a first and a second programming language, and application functions of a first and a second source.

35. A method comprising:

enforcing a first privilege protection with at least two privilege levels when executing programming instructions; and

concurrently, enforcing a second privilege protection with at least two sub-privilege levels for at least one of the privilege level of the first privilege protection.

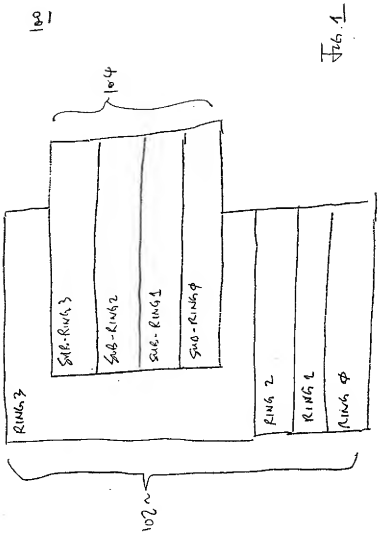
36. The method of claim 35, wherein

said enforcing of the first privilege protection comprises affording the same privilege level of the first privilege protection to a first and a second plurality of sets of programming instructions of a first and a second type, and

said enforcing of the second privilege protection comprises affording a first and a second sub-privilege level of the second privilege protection to the first and second programming instruction sets of the first and second types respectively, the second software facilitated sub-privilege level having less or equal privileges than the first software facilitated sub-privilege level.

37. The method of claim 36, wherein the first and second types are two selected from a group consisting of core system services, programming language runtime supports, and application functions.

38. The method of claim 36, wherein the first and second types are selected from a group consisting of programming language runtime support of a first and a second programming language, and application functions of a first and a second source.



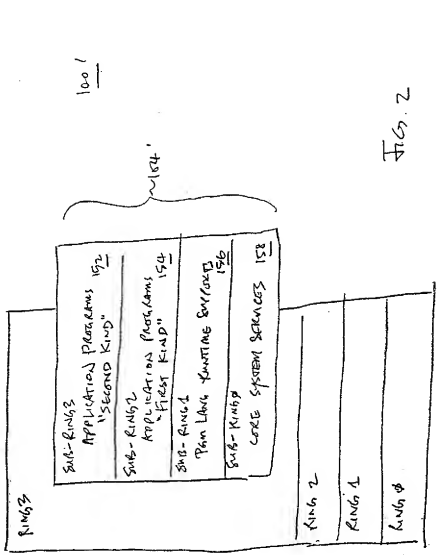


Fig. 2

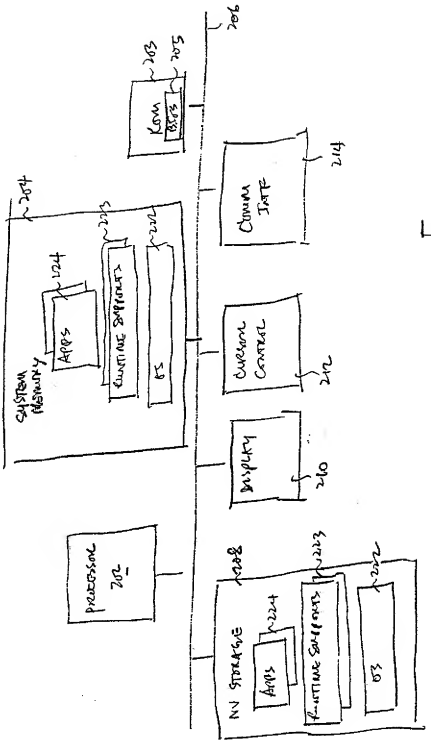

SYSTEM MEMORY
APPS
EXISTING SOFTWARE
OS
PROCESSOR
NV STORAGE
APPS
FUTURE SOFTWARE
OS
BRIDGE
CURSOR CONTROL
COMM INTF
KBD/MS
202
204
206
208
210
212
214
222
224
226
228

Fig 3

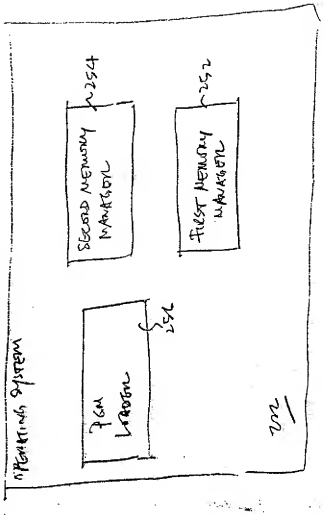


FIG. 4

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/04583

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F12/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, IBM-TDB, COMPENDEX, INSPEC, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 47072 A (HAL COMPUTER SYSTEMS INC) 22 October 1998 (1998-10-22) page 9, line 12 -page 29, line 6; figures 1-19 ---	1-3, 8-18, 23-38
Y	ANONYMOUS: "Two Mode Storage Protection Mechanisms. July 1976." IBM TECHNICAL DISCLOSURE BULLETIN, vol. 19, no. 2, 1 July 1976 (1976-07-01), pages 425-428, XP002168726 New York, US page 427, last paragraph -page 428, last paragraph; figures 4-7 --- -/--	1-38

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
1 June 2001	06/07/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Weber, R

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
Y	EP 0 472 487 A (IBM) 26 February 1992 (1992-02-26) column 5, line 27 -column 9, line 52; figures 2,3,5	1-38
Y	US 5 163 096 A (SINHA BHASKAR ET AL) 10 November 1992 (1992-11-10) column 7, line 31 -column 9, line 15; figures 9,10	1-38

INTERNATIONAL SEARCH REPORT

In: ational Application No
PCT/US 01/04583

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9847072 A	22-10-1998	AU 6549198 A	11-11-1998
EP 0472487 A	26-02-1992	US 5280614 A	18-01-1994
		JP 2001368 C	20-12-1995
		JP 4239354 A	27-08-1992
		JP 7036172 B	19-04-1995
US 5163096 A	10-11-1992	CA 2064640 A,C	07-12-1992
		WO 9222032 A	10-12-1992
		EP 0587587 A	23-03-1994
		HU 67635 A	28-04-1995
		JP 2001383 C	20-12-1995
		JP 5204762 A	13-08-1993
		JP 7036171 B	19-04-1995
		PL 170547 B	31-12-1996
		SK 136193 A	10-08-1994

Form PCT/ISA/210 (patent family annex) (July 1992)